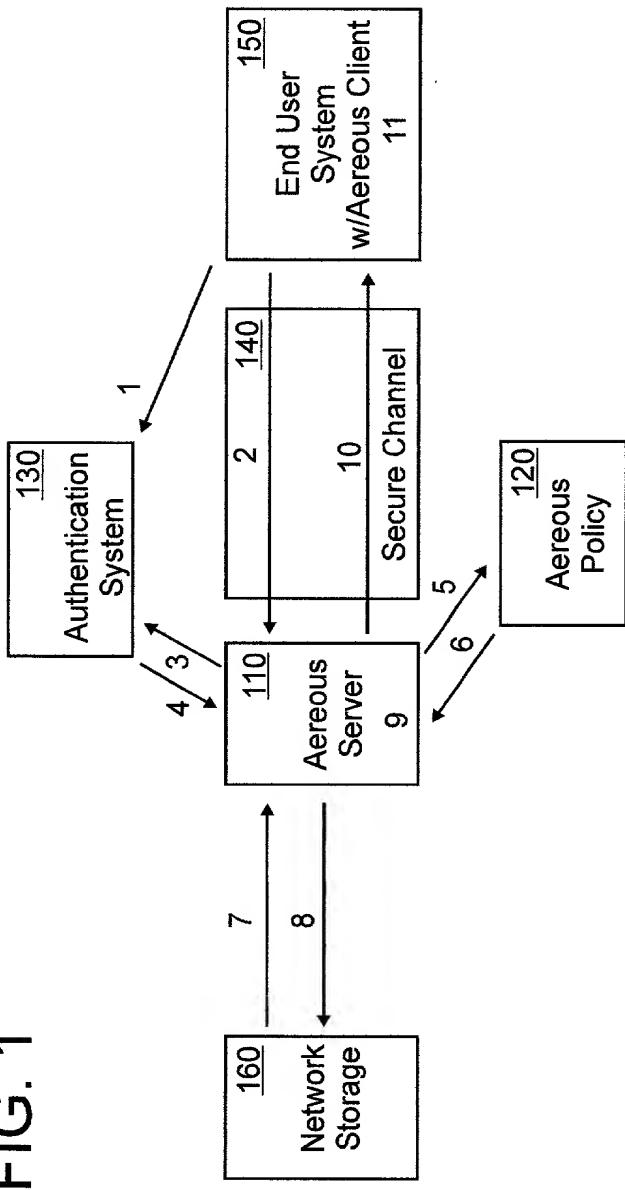


FIG. 1



- 1 Network User logs into the computer network
- 2 Network User contacts the Aereous Server to request a file
- 3,4 Aereous Server contacts the Authentication System to exchange information about the end user.
- 5,6 Aereous Server contacts the Aereous Policy System and executes the access policy to determine if the user has the privilege to access the file. If the answer is yes, then the usage policy is sent to the Aereous Server, if no then the file is not sent to the end user
- 7 Aereous Server requests the file from the network storage device
- 8 Network storage device delivers the file to the Aereous Server
- 9 Aereous Server applies usage rights to, and encrypts the file
- 10 The file is securely delivered to the End User System
- 11 Usage rights and auditing is enforced on the End User System by the Aereous Client

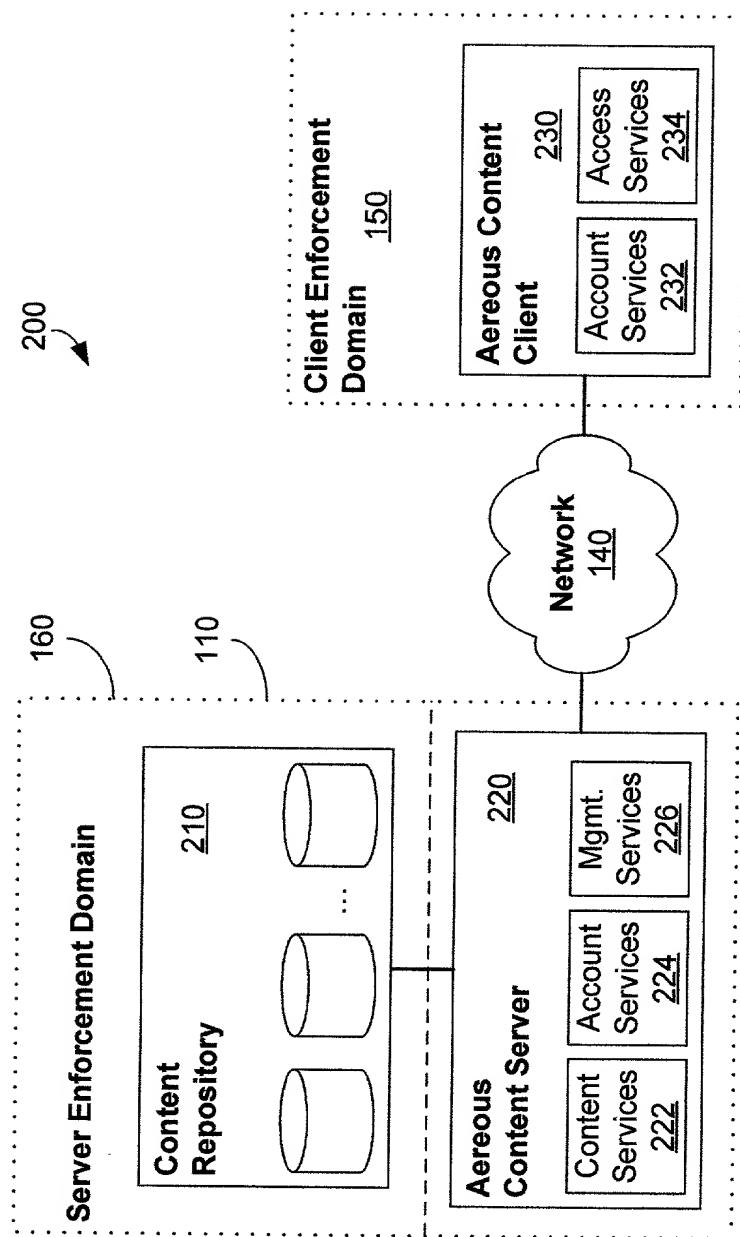


FIG. 2

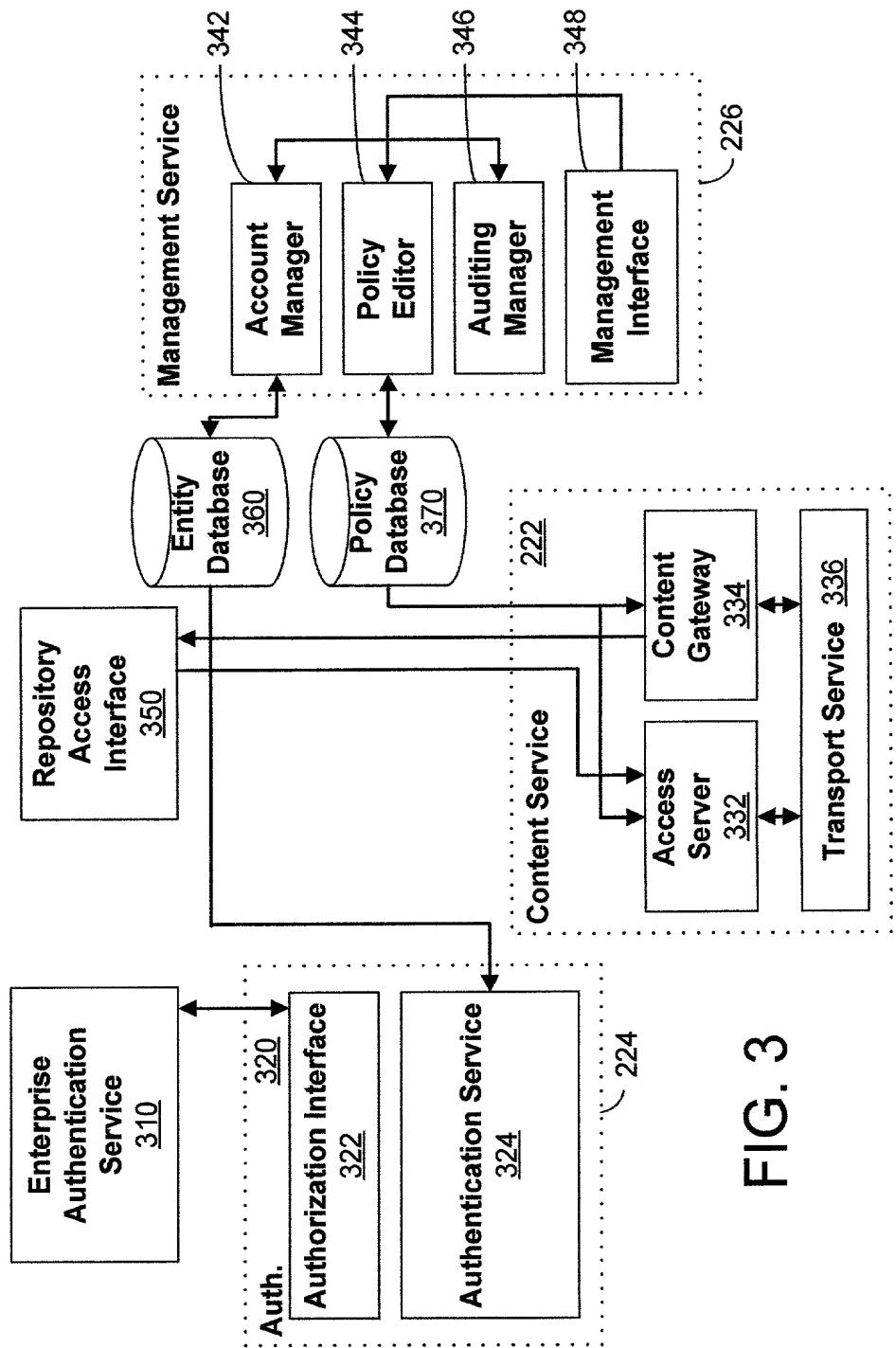


FIG. 3

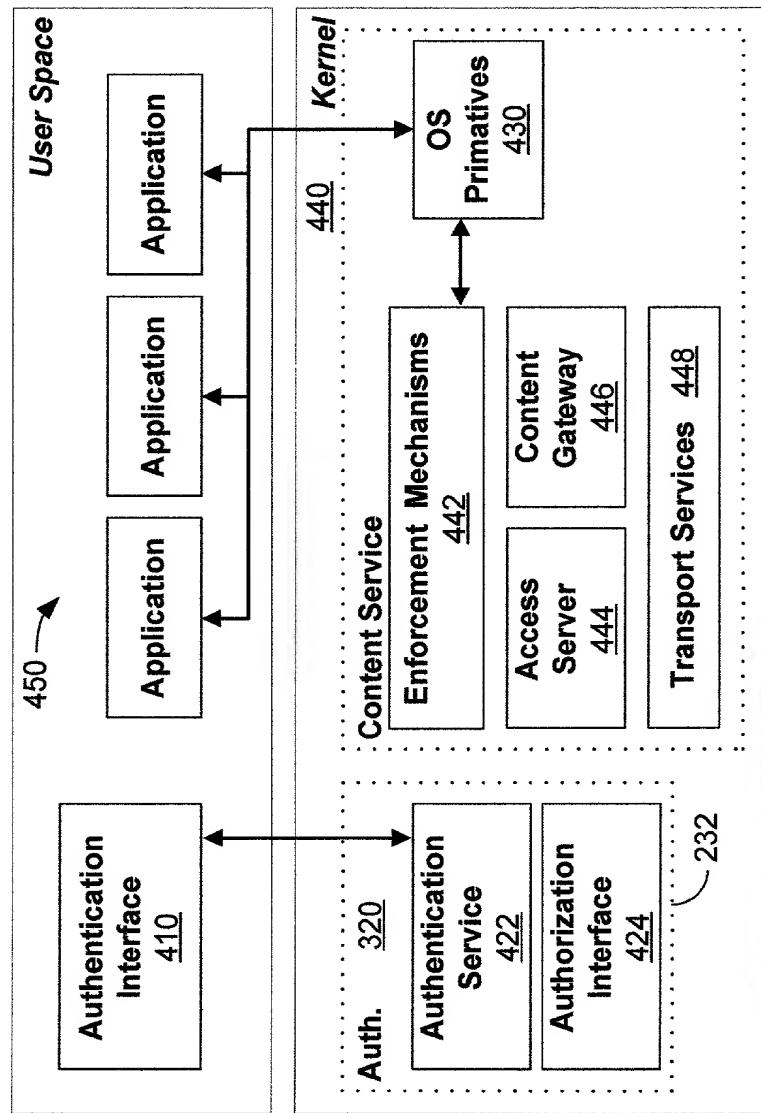
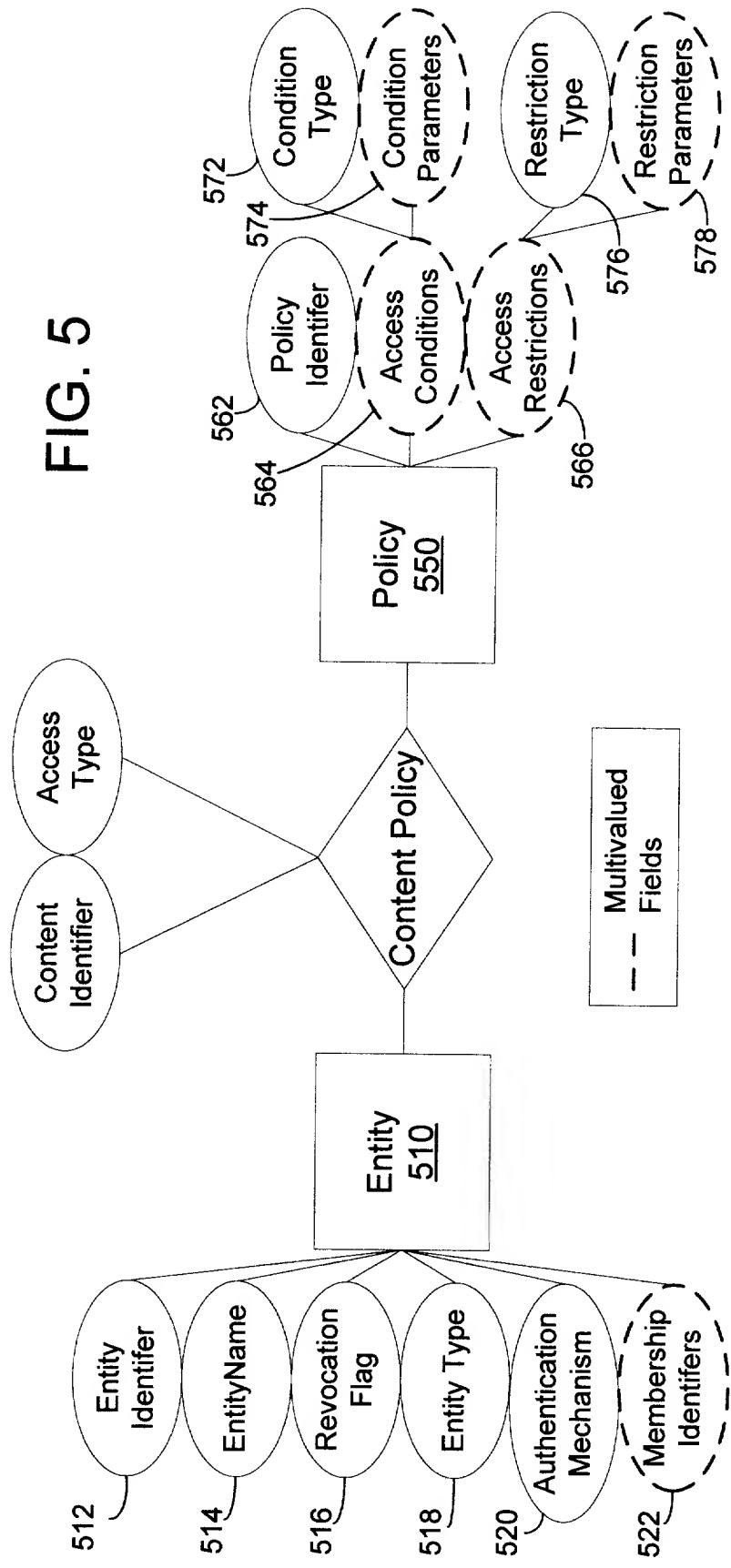


FIG. 4

SCANNED BY

5/23

FIG. 5



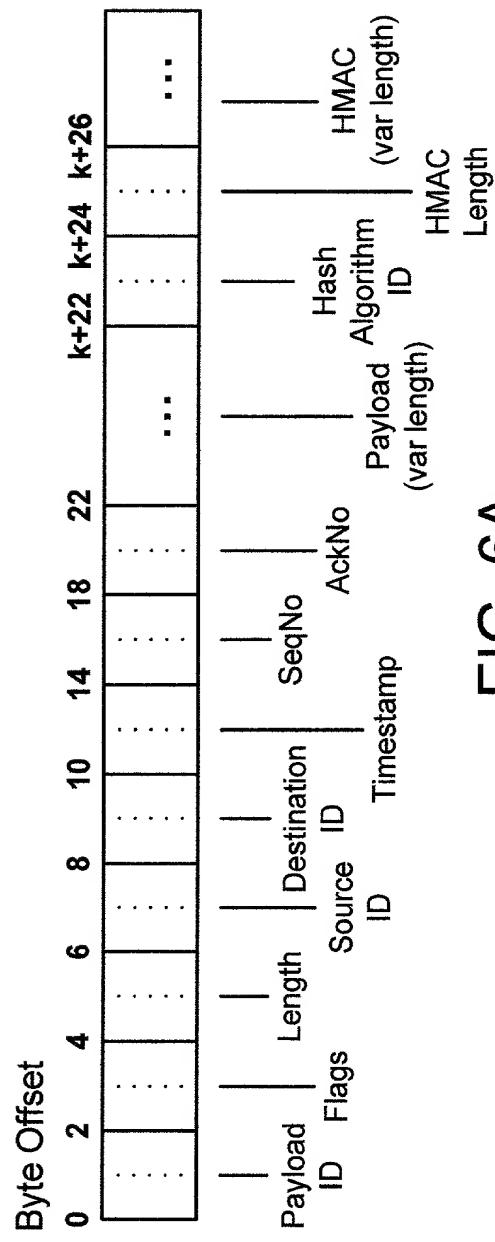


FIG. 6A

Field	Length	Description															
Payload ID	2 bytes	<p>Enumerated type describing the payload type of the message: Further processing of the message is directed by this field. The currently payload identifiers include:</p> <table> <thead> <tr> <th>Type</th><th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>AERE_INVALID</td><td>0</td><td>Invalid type</td></tr> <tr> <td>AERE_FILE_KEY</td><td>1</td><td>File key payload</td></tr> <tr> <td>AERE_BLK_XFER</td><td>2</td><td>Block transfer</td></tr> <tr> <td>AERE_STATUS</td><td>3</td><td>Aereous status</td></tr> </tbody> </table>	Type	Value	Description	AERE_INVALID	0	Invalid type	AERE_FILE_KEY	1	File key payload	AERE_BLK_XFER	2	Block transfer	AERE_STATUS	3	Aereous status
Type	Value	Description															
AERE_INVALID	0	Invalid type															
AERE_FILE_KEY	1	File key payload															
AERE_BLK_XFER	2	Block transfer															
AERE_STATUS	3	Aereous status															
Flags	2 bytes	<p>Flags indicating payload processing requirements. The currently defined flags include:</p> <table> <thead> <tr> <th>Flag</th><th>Bit</th><th>Description</th></tr> </thead> <tbody> <tr> <td>Encrypted</td><td>0</td><td>Payload encrypted</td></tr> <tr> <td>Signed</td><td>1</td><td>Payload signed (not implemented)</td></tr> <tr> <td>Reserved</td><td>2-15</td><td><i>unused</i></td></tr> </tbody> </table>	Flag	Bit	Description	Encrypted	0	Payload encrypted	Signed	1	Payload signed (not implemented)	Reserved	2-15	<i>unused</i>			
Flag	Bit	Description															
Encrypted	0	Payload encrypted															
Signed	1	Payload signed (not implemented)															
Reserved	2-15	<i>unused</i>															
Length	2 bytes	Length of message, in bytes. This length measures the field through the last byte of the payload.															
Source ID	2 bytes	Source identifier – uses user or server entity identifier defined in the entity database.															
Destination ID	2 bytes	Recipient identifier – uses user or server entity identifier defined in the entity database.															
Timestamp	4 bytes	Timestamp (obtained from local or trusted timing source) of message creation. Used to ensure freshness (e.g., mitigate replay attacks). The time is represented by the standard POSIX 32 bit second identifier (seconds since epoch).															
SeqNo	2 bytes	Sequence number used to ensure the ordering of messages.															
AckNo	2 bytes	Acknowledgement of all messages up to including <i>Ackno</i> .															
Payload	variable	This is the variable length data to be interpreted by payload processing. The format of the payload is detailed in Section 7.3. Based on message flags, this data require additional process (e.g., encryption, sign).															
Hash Algo. Identifier	2 bytes	<p>Enumerate type defining the hash algorithm used in the calculation of the keyed hash. The following hash algorithms are supported by the Aereous system;</p> <table> <thead> <tr> <th>Algorithm</th><th>Value</th></tr> </thead> <tbody> <tr> <td>AERE_MD5</td><td>0</td></tr> <tr> <td>AERE_SHA1</td><td>1</td></tr> </tbody> </table>	Algorithm	Value	AERE_MD5	0	AERE_SHA1	1									
Algorithm	Value																
AERE_MD5	0																
AERE_SHA1	1																
HMAC Length	2 bytes	The length of the HMAC value. Note that some cryptographic algorithms output more ciphertext than the original plaintext. (Question: Is this really needed, or can we always calculate this from the key/hash algorithm info?)															
HMAC	variable	This is the keyed hash of the message. This value is calculated over all bytes prior to the beginning of the hash length field.															

FIG. 6B

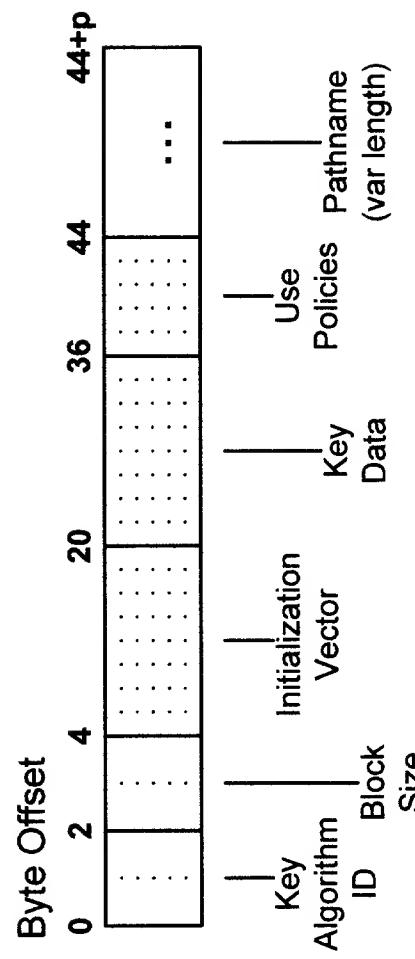


FIG. 7

Name	Length	Description															
KeyAlgorithmID	16 bits	(enumerated) identifies both the algorithm and the key length															
BlockSize	16 bits	block size for the accessed file															
IV	256 bits	Initialization vector used to seed the encryption of file blocks. Further details are defined in Section 7.1.															
KeyData	256 bits	The key used to encrypt the file. Where the key size is less than 256 bits, the most significant bits are used and unused bits are padded with zero.															
UsePolicies	64 bits	Flags indicating the enabled usage of access content (where a bit 1 = allowed, 0 = denied). The supported bits include: <table> <thead> <tr> <th>Flag</th> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Print</td> <td>0</td> <td>Print the file</td> </tr> <tr> <td>Copy</td> <td>1</td> <td>Copy file to local disk</td> </tr> <tr> <td>Send</td> <td>2</td> <td>Transmit the file to external device</td> </tr> <tr> <td>Reserved</td> <td>3-63</td> <td>unused</td> </tr> </tbody> </table> <p><i>NOTE: The set of usage types are identified in the Aereous Client Design Document, and will be reflected in future version of this document as needed.</i></p>	Flag	Bit	Description	Print	0	Print the file	Copy	1	Copy file to local disk	Send	2	Transmit the file to external device	Reserved	3-63	unused
Flag	Bit	Description															
Print	0	Print the file															
Copy	1	Copy file to local disk															
Send	2	Transmit the file to external device															
Reserved	3-63	unused															
Pathname	(variable)	full pathname of file being accessed															

FIG. 8

Name	Length	Description
Cid	16 bits	hashed pathname identifier (see Section 6)
Block/Number	16 bits	block number of transmitted data
Length	16 bits	length of data, typically equal to the block size supported by the filesystem
Data	(variable)	the file data

FIG. 9

Name	Length	Description																																			
Sid	16 bits	<p>(enumerated) Type identifying the message semantics.  <i>Details of the status are further specified in the info and text fields.</i></p> <table> <thead> <tr> <th>Enum</th><th>Numeric</th><th>Origin</th><th>Description</th></tr> </thead> <tbody> <tr> <td>usageExec</td><td>0</td><td>client</td><td>Usage right executed</td></tr> <tr> <td>aereousError</td><td>1</td><td>both</td><td>Aereous error encountered</td></tr> <tr> <td>dfsError</td><td>2</td><td>both</td><td>Filesystem error</td></tr> <tr> <td>infoStatus</td><td>3</td><td>both</td><td>informational (e.g., debugging)</td></tr> <tr> <td>clientShutdown</td><td>4</td><td>client</td><td>client shutdown signal</td></tr> <tr> <td>serverShutdown</td><td>5</td><td>server</td><td>server shutdown signal</td></tr> <tr> <td><i>unused</i></td><td>6-2--</td><td>N/A</td><td><i>unused</i></td></tr> </tbody> </table>				Enum	Numeric	Origin	Description	usageExec	0	client	Usage right executed	aereousError	1	both	Aereous error encountered	dfsError	2	both	Filesystem error	infoStatus	3	both	informational (e.g., debugging)	clientShutdown	4	client	client shutdown signal	serverShutdown	5	server	server shutdown signal	<i>unused</i>	6-2--	N/A	<i>unused</i>
Enum	Numeric	Origin	Description																																		
usageExec	0	client	Usage right executed																																		
aereousError	1	both	Aereous error encountered																																		
dfsError	2	both	Filesystem error																																		
infoStatus	3	both	informational (e.g., debugging)																																		
clientShutdown	4	client	client shutdown signal																																		
serverShutdown	5	server	server shutdown signal																																		
<i>unused</i>	6-2--	N/A	<i>unused</i>																																		
Infolength	16 bits	length of <i>info</i> field																																			
Info	(variable)	Additional status information. The interpretation of this field is directed by the <i>Sid</i> field as follows:																																			
		<table> <thead> <tr> <th>Enum</th><th>Subfields</th></tr> </thead> <tbody> <tr> <td>usageExec</td><td>content ID (<i>cid</i>), usage mask</td></tr> <tr> <td>aereousError</td><td>Aereous error code</td></tr> <tr> <td>dfsError</td><td>standard UNIX error</td></tr> <tr> <td>infoStatus</td><td>information enum</td></tr> <tr> <td>clientShutdown</td><td><i>none</i></td></tr> <tr> <td>serverShutdown</td><td><i>none</i></td></tr> <tr> <td><i>unused</i></td><td><i>unused</i></td></tr> </tbody> </table>				Enum	Subfields	usageExec	content ID ( <i>cid</i> ), usage mask	aereousError	Aereous error code	dfsError	standard UNIX error	infoStatus	information enum	clientShutdown	<i>none</i>	serverShutdown	<i>none</i>	<i>unused</i>	<i>unused</i>																
Enum	Subfields																																				
usageExec	content ID ( <i>cid</i> ), usage mask																																				
aereousError	Aereous error code																																				
dfsError	standard UNIX error																																				
infoStatus	information enum																																				
clientShutdown	<i>none</i>																																				
serverShutdown	<i>none</i>																																				
<i>unused</i>	<i>unused</i>																																				
TextLength	16 bits	length of <i>Text</i> field																																			
Text	(variable)	C-string description of information. Used in auditing or as user notification.																																			

FIG. 10

FIG. 11

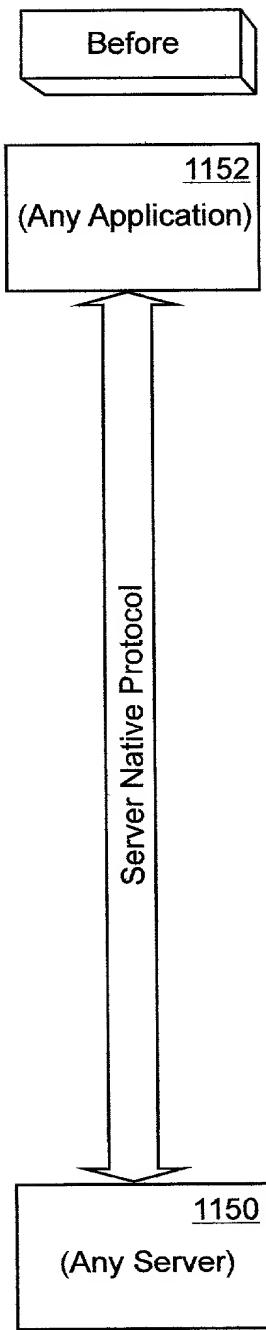
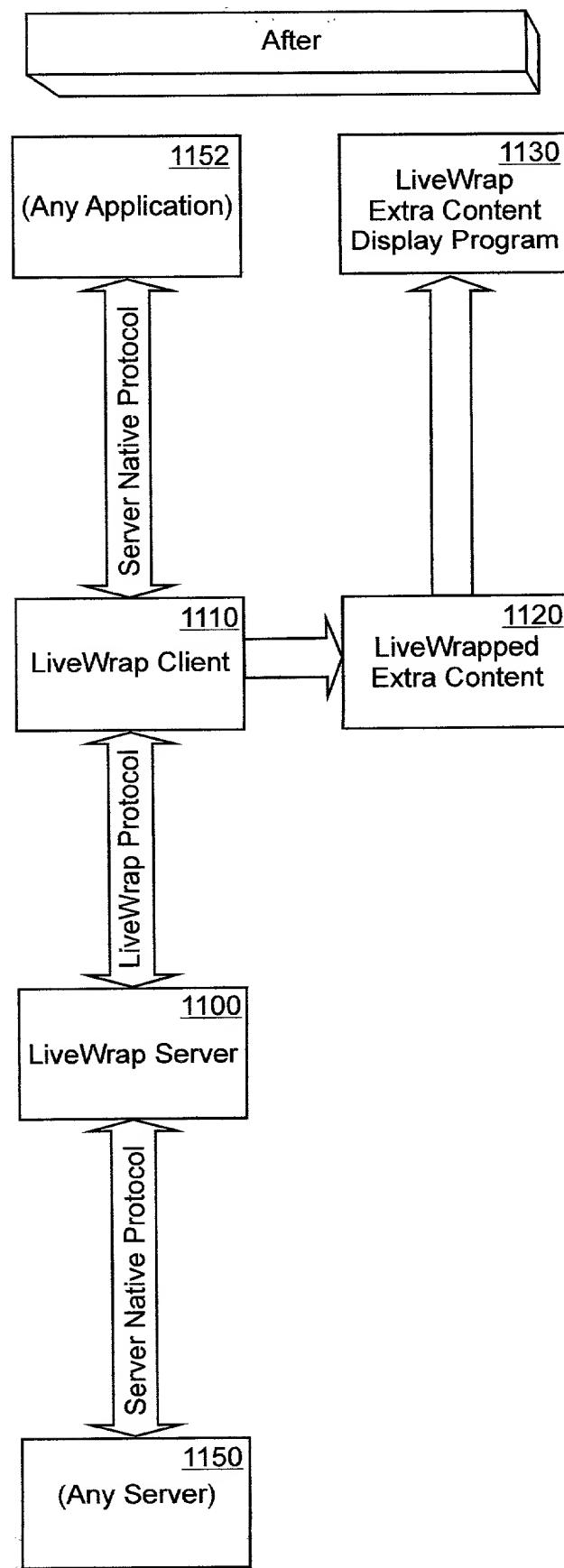


FIG. 12



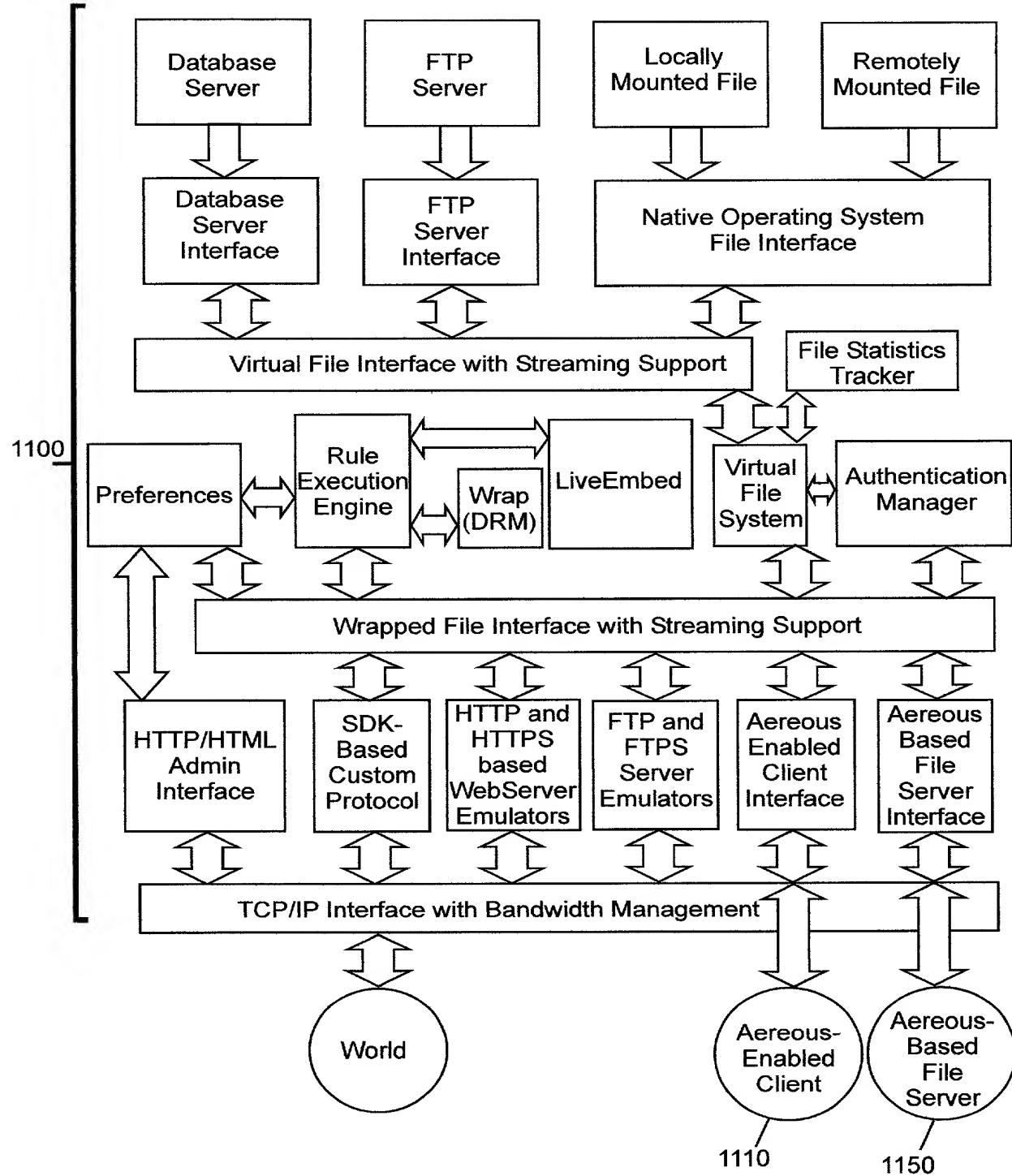


FIG. 13

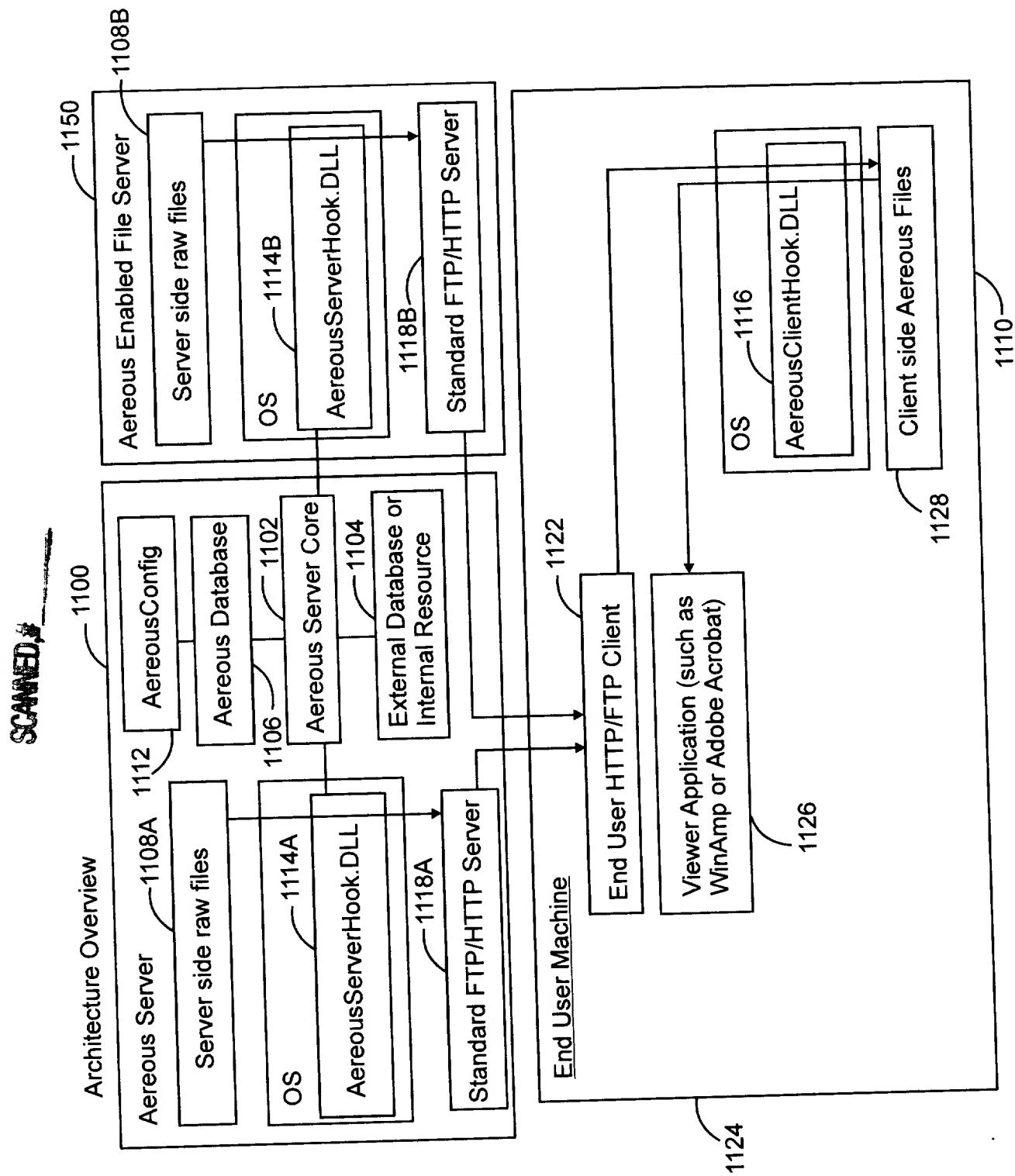


FIG. 14

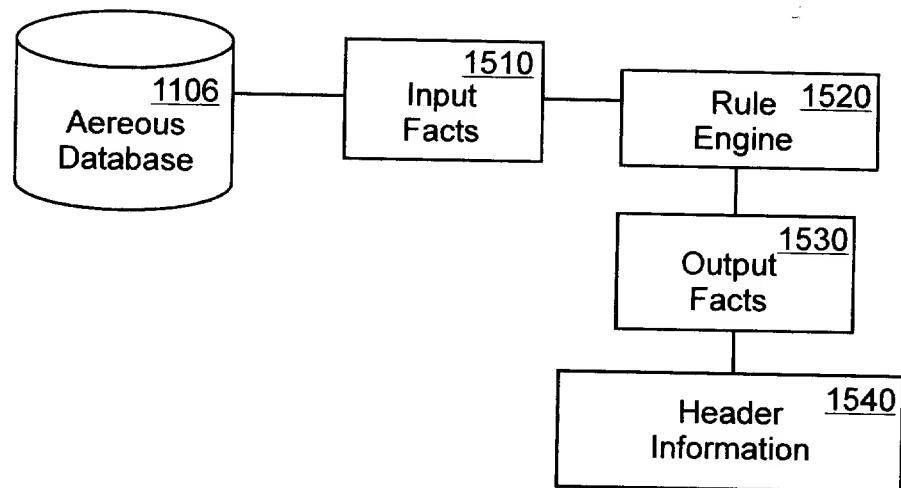


FIG. 15

SCANNED BY

## AereousServer Core Plug-in Architecture

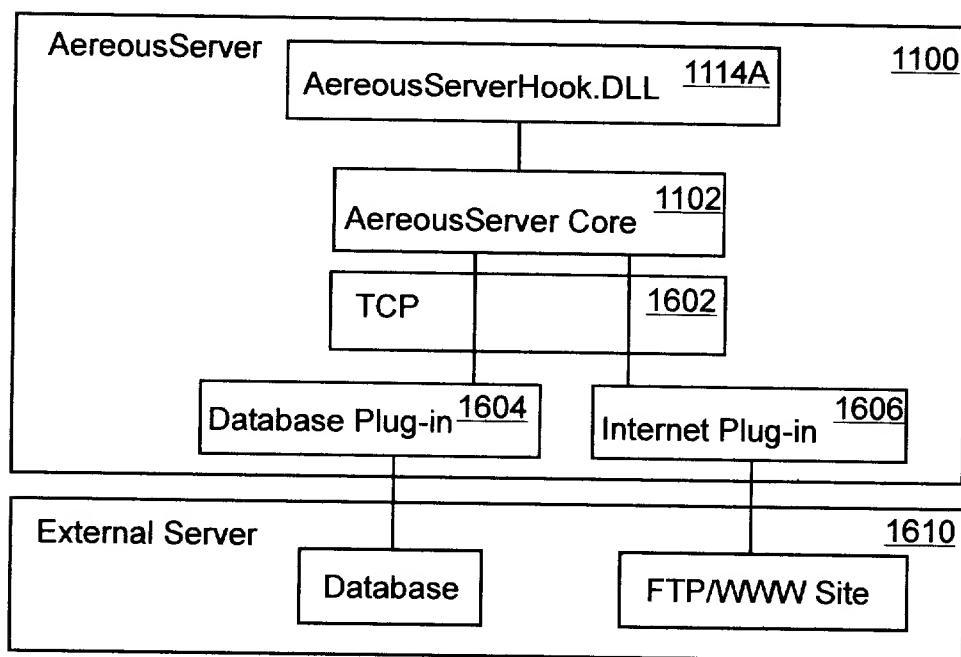


FIG. 16

File Server Example Interaction Diagram

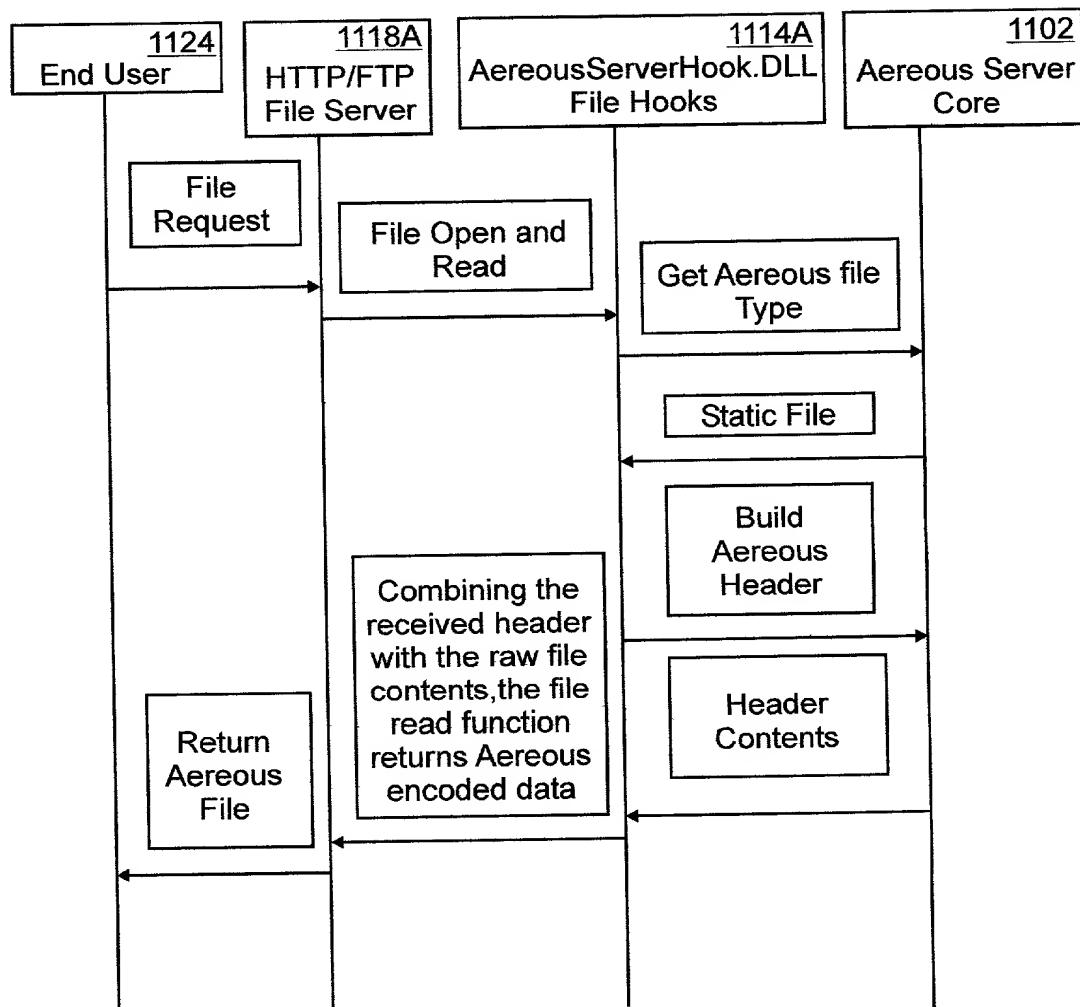


FIG. 17

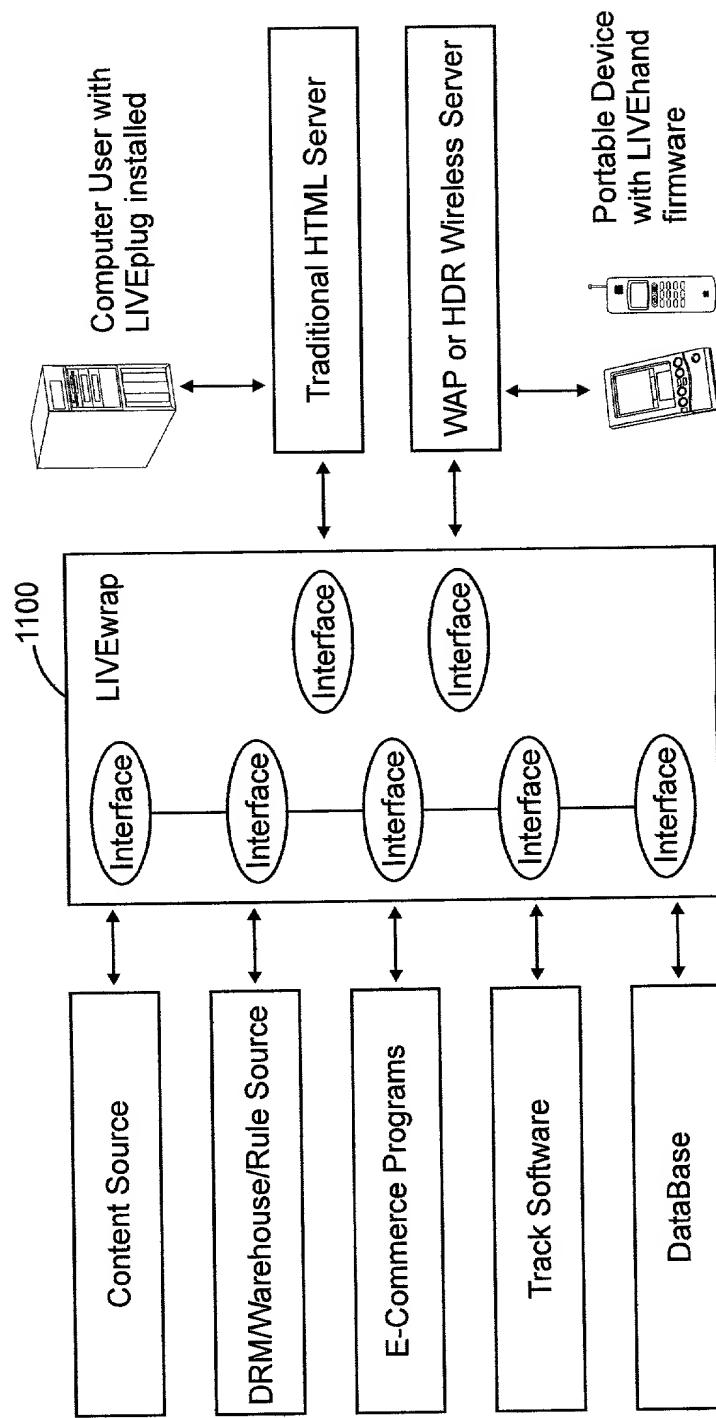


FIG. 18

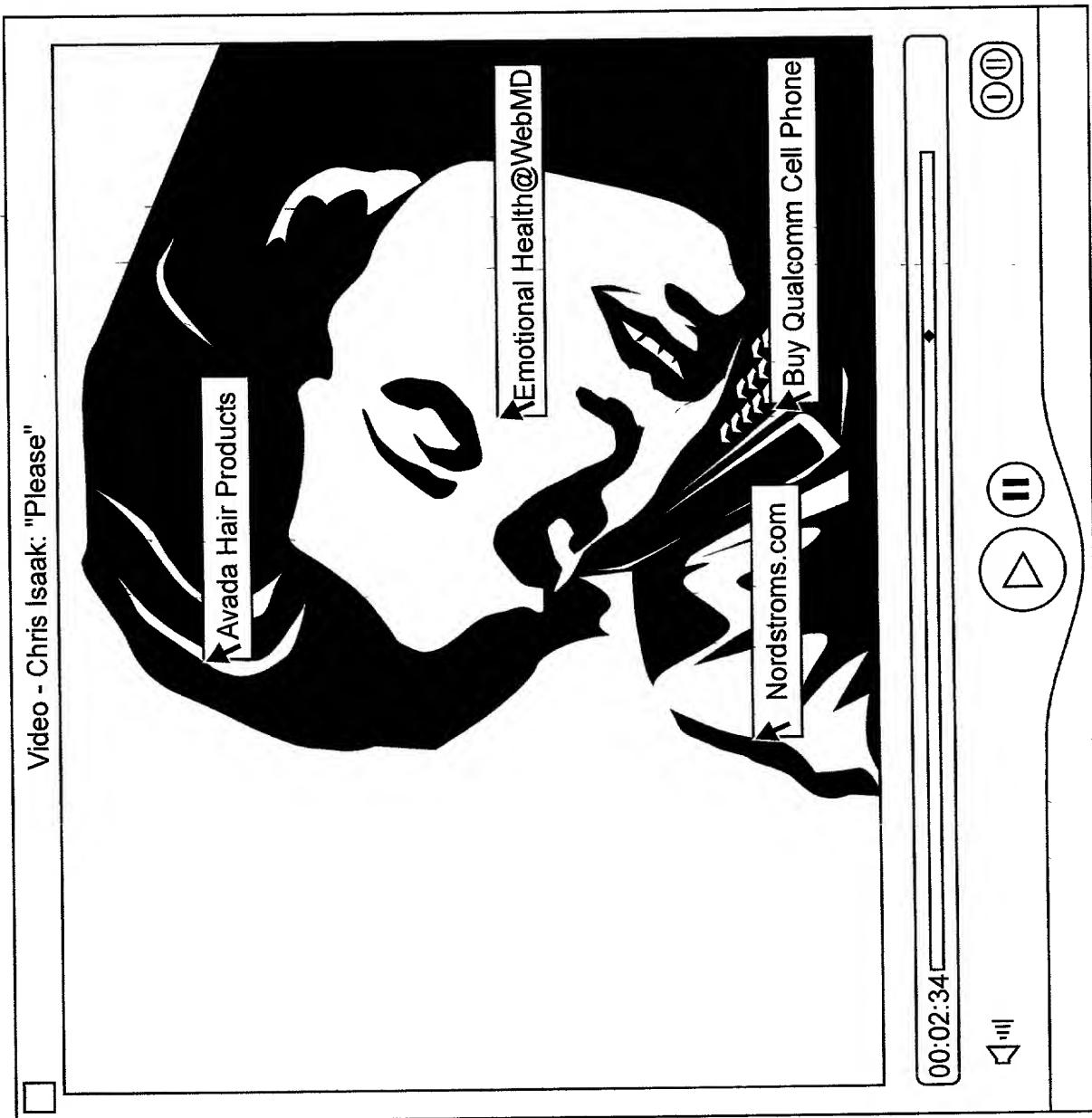


FIG. 19

Start: 7/30/00 11:49:07 PM  
 PING from: 24.160.215.100  
 to: www. .com

PKT#	RESULT	TIME(ms)	LENGTH	Every 500 mSecs			Timestamp
				Min	Avg	Max	
1	success	176	40	176	176	176	7/30/00 11:49:07 PM
2	success	134	40	134	155	176	7/30/00 11:49:08 PM

Packets out/in/bad/loss = 2/2/0/0  
 Round Trip Time (ms) min/avg/max = 134/155/176  
 • Ping Completed 7/30/00 11:49:08 PM •

FIG. 20

Description	Size (bytes)	Contents
Aereous Signature	11	'AEREOUS' + 0x01301976
Aereous File Version	1	Currently 0x1
File ID	8	File's Aereous ID.
Usage Count	2	Number of usages remaining. Set to 0xFFFF for infinite usages.
Expiration Date	4	A GMT ANSI RTL style time date stamp that indicates when this file expires.
Usage Denied Content	Varying	Once a read attempt fails due to a 0 usage count, this content is displayed to the user. The format is described below under "Content Format."
Number of Push Content Items	2	Number of items that are pushed to the user when the file is opened.
Push Content Items	Varying	Array of push content items. The format is described below as "Push Content Item Format."
Header CRC	4	A CRC value for the preceding header bytes.
Content Size	8	The size of the unencrypted data
Encryption Type	1	0 = Unencrypted 1 = 2Fish 2-255 = undefined
Encrypted Data Offset	8	A file offset to the beginning of the encrypted data. The encrypted data uses the format described in "Encrypted Data Block."

FIG. 21

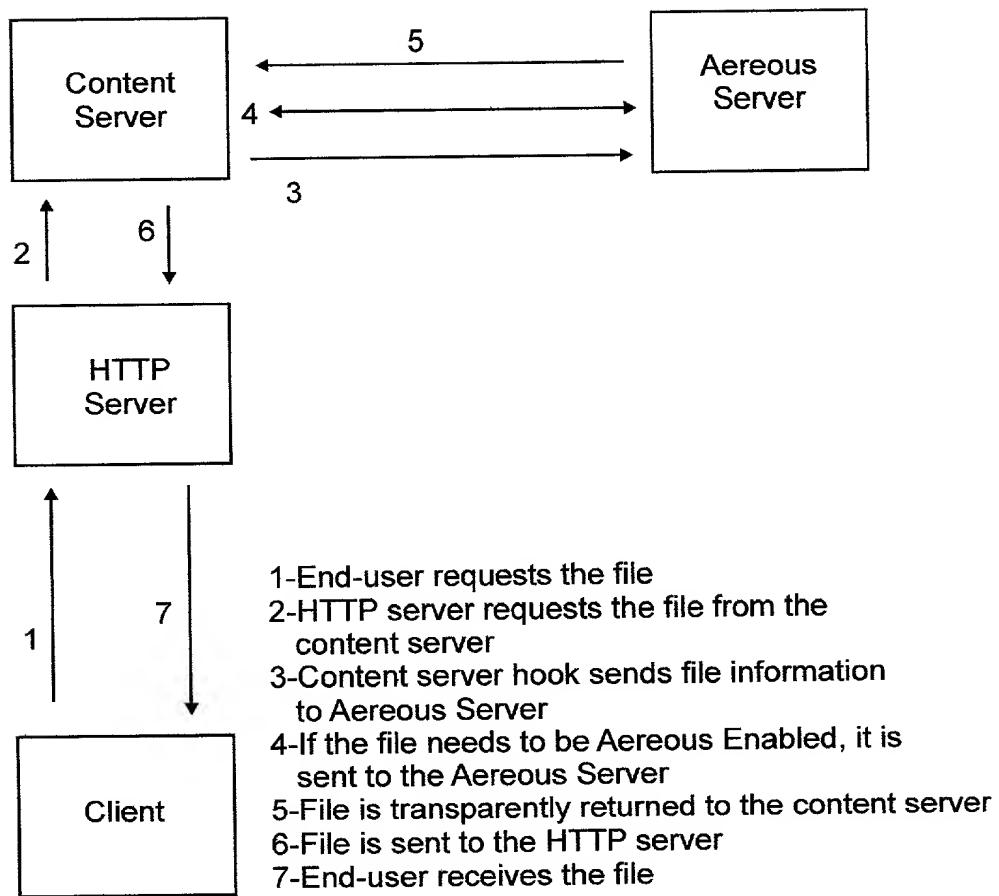


FIG. 22

**VirtualFile**

The VirtualFile table lists all files in the system. Each file is associated with a Plug-In and a bundle of facts that are understandable by that Plug-In.

Column	Type	Description	Sample
*VirtualFile ID	Int	System assigned ID	314
Name	Text	Name of the virtual file. This is the base name, with no parent directory names and no directory separator characters. The name is not case sensitive in the server core, but is allowed to be in the database engine.	SalesReport.doc
IsDefault	Bool	Flag indicating whether Name is actually a wildcard pattern match. Use of this flag allows directories to be setup and facts associated with them without having to database each of the files that could reside within that virtual directory.	False
VirtualDirectoryID (optional)	Int	ID of the VirtualDirectory that the file resides within. Use NULL for files that reside at the root level.	4242
PlugInName	Text	Identifies which plug-in will generate the actual file contents.	FTP
IsStatic	Bool	Flag indicating whether the file is an actual static file on the server disk of a true virtual file.	True
FactBundleID (optional)	Int	Facts for this file. These facts are considered to be "owned" by this file and will be deleted if this file entry is deleted.	4243
SharedFactBundleID (optional)	Int	Facts for this file. These facts are not "owned" by this file, instead existing as shared facts to assist with centralized administration.	12000
ShouldLogUsageEvents	Int	Flag indicating whether any access to this file should result in an access log. <ul style="list-style-type: none"> <li>• 1 indicates there should be a log generated</li> <li>• 0 means no log should be generated</li> <li>• Null or -1 means that the value of this setting should be inherited from the parent directory or the DefaultShouldLogUsageEvents configurable parameter</li> </ul>	1

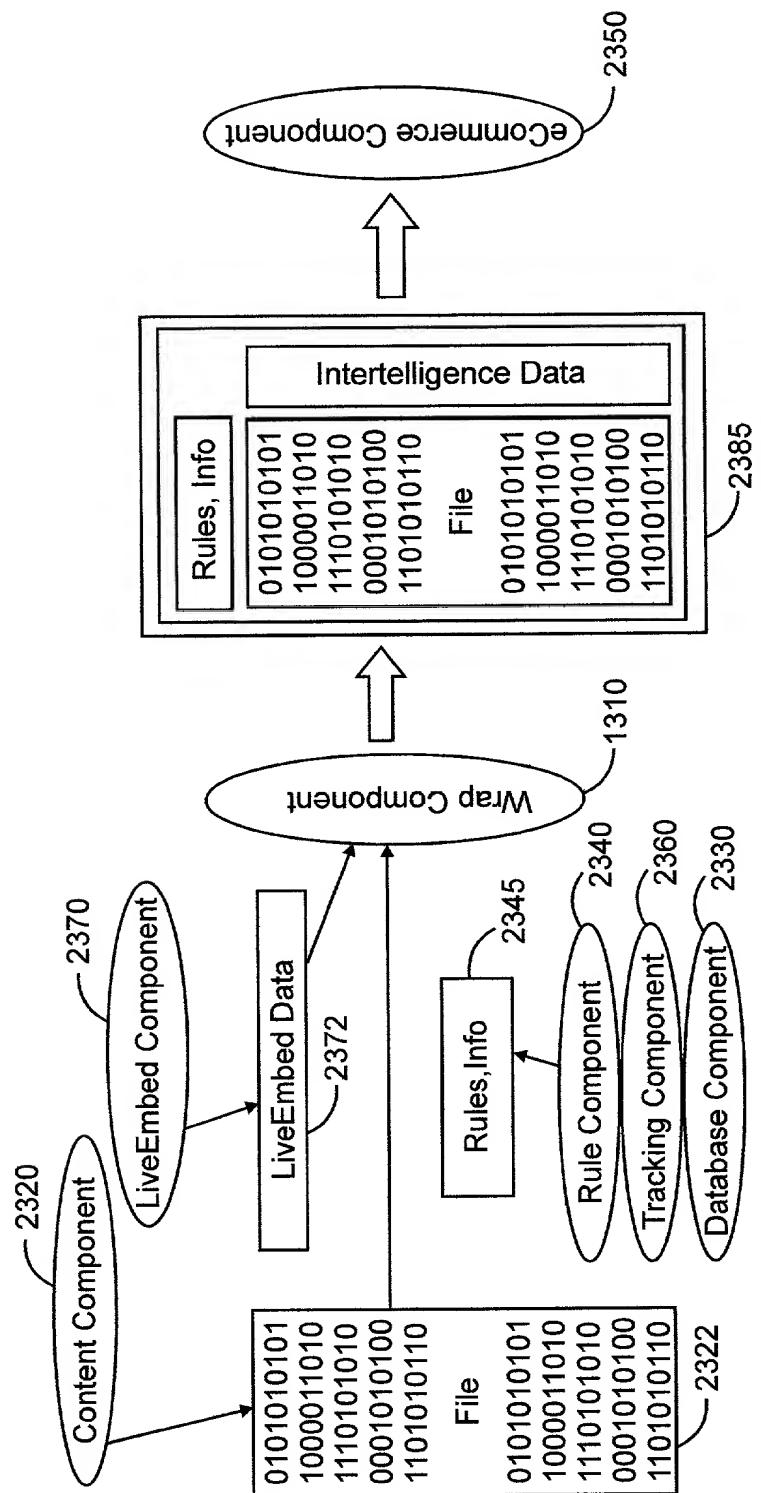
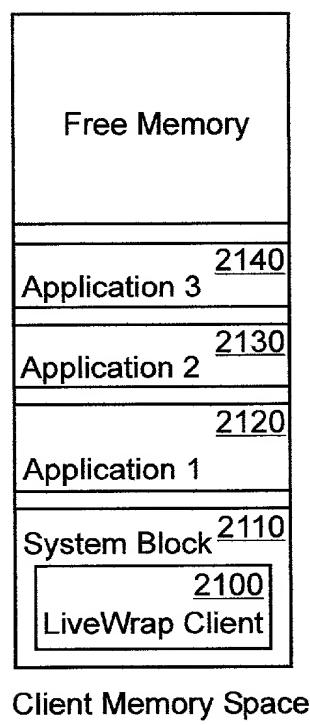


FIG. 24



Client Memory Space

FIG. 25